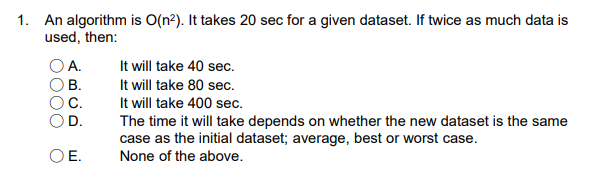
100 – 14 – 12 = 74

Multiple Choice



D

A picture containing text, font, algebra, screenshot

Description automatically generated

E

A picture containing text, screenshot, font

Description automatically generated

C

A white background with black text

Description automatically generated with low confidence

D

A picture containing text, font, algebra, screenshot

Description automatically generated

E

A white background with black text

Description automatically generated with low confidence

B

A picture containing text, font, algebra, screenshot

Description automatically generated

C

A picture containing text, screenshot, font, algebra

Description automatically generated

D E

A white background with black text

Description automatically generated with low confidence

D

A picture containing text, screenshot, font

Description automatically generated

A D

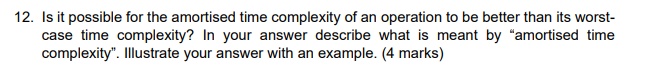
Short Answer

A white background with black text

Description automatically generated with low confidence

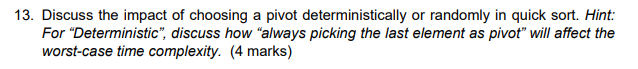
T(n) = 2\*2T(n-2) + O(1)

guess O(2^n)



Amortised time complexity is that normal time complexity is better than its worst-case time complexity. Insertion, O(n), half sort O(log(n))

If you average out the worst case time complexity of the operation (O(n)) over the total number of operations performed before an array resize occurs (n operations, as the array is only resized when full), then the amortised time complexity is O(1). This is less than the worst case time complexity of O(n) for insert. -2



if the original array is sorted by descending order, always picking the last element as pivot will have O(n^2) time complexity. Every cycle, it will transverse over all elements.

-2

A picture containing text, font, screenshot

Description automatically generated

a 0

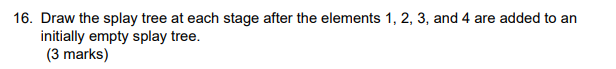
b Ω(n)

A picture containing text, font, screenshot, line

Description automatically generated

array, current node is i, new last node is i+1. O(1).

linked list, transverse parents and uncles of current node, if left child, insert in right child. if two children, pass, if no child, left child. than, if there is no available position, insert the node is left most brother’s left child. O(logn)



1

2

3

4

-3

A picture containing text, font, line, screenshot

Description automatically generated

new hash table

hash.add(0, head.next)

for (i=1, i<n, i++){

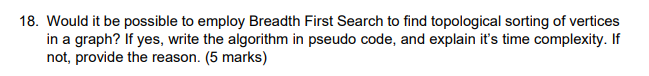
if hash.get(i-1).next == hash.get(0)

return true

}

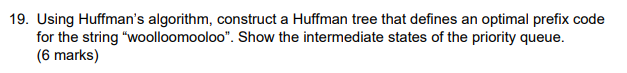
return false

-3



yes

-4



w 1

o 8

l 3

m 1

5 8

2 3

1 1

o 01

l 001

w 0000

m 0001